A MULTIDIMENSIONAL FINITE ELEMENT METHOD FOR CFD

Darrell W. Pepper and Joseph W. Humphrey
Advanced Projects Research, Inc.
5301 N. Commerce Ave.
Moorpark, CA 93021

ABSTRACT

A finite element method is used to solve the equations of motion for 2- and 3-D fluid flow. The time-dependent equations are solved explicitly using quadrilateral (2-D) and hexahedral (3-D) elements, mass lumping, and reduced integration (when appropriate). A Petrov-Galerkin technique is applied to the advection terms. The method requires a minimum of computational storage, executes quickly, and is scalable for execution on computer systems ranging from PCs to supercomputers.

NOMENCLATURE

A	global advection matrix
$a_{i,j}$	local advection matrix
C	gradient matrix
C^T	divergence matrix
E_T	total energy
F_u	body force term for x momentum
F_v	body force term for y momentum
F_w	body force term for z momentum
F_E	load vector for energy equation
$k_{i,j}$	local diffusion matrix
K	global diffusion matrix
$m_{i,j}$	local mass matrix
M	global mass matrix
M_∞	Mach number
N	basis function
p	pressure
P_r	Prandtl number
R_a	Rayleigh number
R_e	Reynolds number
S	boundary segment
T	temperature
t	time
u	horizontal velocity
$\bar{u}$	velocity vector
U_∞	reference velocity
v	lateral velocity
w	vertical velocity
x	horizontal coordinate
y	lateral coordinate
z	vertical coordinate

W weighting function
α Petrov-Galerkin parameter
β Petrov weighting factor
γ ratio of specific heats
Δt time step
κ thermal diffusivity
ρ density
σ_{xx} x-coordinate viscous stress
σ_{xy} x-y viscous shear
σ_{xz} x-z viscous shear
σ_{yy} y-coordinate viscous stress
σ_{yz} y-z viscous shear
σ_{zz} z-coordinate viscous stress
Ω area (2-D); volume (3-D)

Subscripts

i,j,k local node index
x x direction
y y direction
z z direction
∞ free stream
_ column vector

Superscripts

T transpose
-1 inverse
n previous time value
n+1 new time value
x x direction
y y direction
z z direction
$\hat{u}$ trial value
$\tilde{u}$ average value

INTRODUCTION

Numerical modeling of multidimensional fluid flow is a difficult, computationally intensive effort. When the physical domain is simple, finite difference methods (FDM) are typically used as a matter of convenience and historical preference. If the domain is irregular, boundary fitted coordinates are usually employed [1]; however, such techniques are not easy to use, particularly when generating 3-D grids, and require the transformation of the governing equations. Finite element methods (FEM), while mathematically more robust and convenient to use in complex geometries, can tax the limits of the largest computers. In an effort to alleviate these constraints, a modified FEM, which is competitive with the FDM in storage and speed, is used to solve a set of example problems in 2- and 3-D for both incompressible and compressible flows. The FEM is based on the method discussed in Pepper and Singer [2] and Pepper and Humphrey [3].

The modified FEM employs simple, well known alterations to the conventional FEM: mass lumping, reduced integration, and application of Petrov-Galerkin weighting. Lumping the time derivative term into one diagonal allows simple explicit time integration to be used and the mass matrix to be easily inverted. One point quadrature, as opposed to 2x2 (bilinear quadrilateral) or 2x2x2 (hexahedral) Gauss points, yields integral values obtained at the element centroid. This procedure permits explicit evaluation of the integral terms on a local level, i.e., per element without regard for bandwidth. The application of reduced quadrature has been found to be very effective when the element is not too distorted [4]; however, care must be exercised in generating the initial mesh to insure aspect ratios and distortions are minimal.

Velocities within the advection terms are averaged over an element and a Petrov-Galerkin formulation applied [5]. Application of the Petrov-Galerkin technique helps maintain stability but does not overly dampen the solution, compared to the more commonly utilized upwinding techniques and anti-dispersive schemes applied in FDMs [6].

Performance of the algorithm on current scalar/vector machines is good [2]; many multidimensional problems can be run on low-end computers. The performance of the FEM is significantly improved on a parallel computer. Since the method is explicit in time, vectorization and parallelization are relatively easy. A version of the method has been written and optimized for execution on an Alliant VFX/40 mini-supercomputer with 4 processors.

GOVERNING EQUATIONS

The mathematical relations which describe the transport of fluid motion and energy are the equations for conservation of mass, momentum, and energy. The governing equations for three-dimensional flow can be written in non-dimensional form as [6]

Conservation of Mass

$$\frac{\partial \rho}{\partial t}+\frac{\partial \rho u}{\partial x}+\frac{\partial \rho v}{\partial y}+\frac{\partial \rho w}{\partial z}=0 \tag{1}$$

Conservation of Momentum

x-direction

$$\rho\frac{\partial u}{\partial t}+\rho u\frac{\partial u}{\partial x}+\rho v\frac{\partial u}{\partial y}+\rho w\frac{\partial u}{\partial z}=-\frac{\partial p}{\partial x}+\frac{\partial \sigma_{xx}}{\partial x}+\frac{\partial \sigma_{xy}}{\partial y}+\frac{\partial \sigma_{xz}}{\partial z} \tag{2}$$

y-direction

$$\rho\frac{\partial v}{\partial t}+\rho u\frac{\partial v}{\partial x}+\rho v\frac{\partial v}{\partial y}+\rho w\frac{\partial v}{\partial z}=-\frac{\partial p}{\partial y}+\frac{\partial \sigma_{xy}}{\partial x}+\frac{\partial \sigma_{yy}}{\partial y}+\frac{\partial \sigma_{yz}}{\partial z} \tag{3}$$

z-direction

$$\rho\frac{\partial w}{\partial t}+\rho u\frac{\partial w}{\partial x}+\rho v\frac{\partial w}{\partial y}+\rho w\frac{\partial w}{\partial z}=-\frac{\partial p}{\partial z}+\frac{\partial \sigma_{xz}}{\partial x}+\frac{\partial \sigma_{yz}}{\partial y}+\frac{\partial \sigma_{zz}}{\partial z} \tag{4}$$

Conservation of Energy

$$\frac{\partial E_t}{\partial t}+\frac{\partial (E_t+p)u}{\partial x}+\frac{\partial (E_t+p)v}{\partial y}+\frac{\partial (E_t+p)w}{\partial z}=\frac{\partial}{\partial x}(u\sigma_{xx}+v\sigma_{xy}+w\sigma_{xz}-q_x)$$

$$+\frac{\partial}{\partial y}(u\sigma_{xy}+v\sigma_{yy}+w\sigma_{yz}-q_y)+\frac{\partial}{\partial z}(u\sigma_{zx}+v\sigma_{zy}+w\sigma_{zz}-q_z) \qquad (5)$$

where ρ is density, u is horizontal velocity, v is lateral velocity, w is vertical velocity, p is pressure, E_t is total energy, q_x, q_y, and q_z are the gradient flux terms for heat conduction, and $\sigma_{xx}, \sigma_{xy}, \sigma_{xz}$, etc., are the normal and tangential viscous stress terms,

$$\sigma_{xx}=\frac{2}{3R_e}\left(2\frac{\partial u}{\partial x}-\frac{\partial v}{\partial y}-\frac{\partial w}{\partial z}\right), \sigma_{yy}=\frac{2}{3R_e}\left(2\frac{\partial v}{\partial y}-\frac{\partial u}{\partial x}-\frac{\partial w}{\partial z}\right), \sigma_{zz}=\frac{2}{3R_e}\left(2\frac{\partial w}{\partial z}-\frac{\partial u}{\partial x}-\frac{\partial v}{\partial y}\right) \qquad (6)$$

$$\sigma_{xy}=\frac{1}{R_e}\left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right), \quad \sigma_{xz}=\frac{1}{R_e}\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right), \quad \sigma_{yz}=\frac{1}{R_e}\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right) \qquad (7)$$

where $R_e=\rho_\infty U_\infty L/\mu_\infty$. For compressible flow, pressure and temperature are obtained from the equation of state

$$p=(\gamma-1)\rho e \qquad (9)$$

$$T=\frac{\gamma M_\infty^2 p}{\rho} \qquad (10)$$

where e is internal energy, $M_{inf}=V_\infty/\sqrt{\gamma R T_\infty}$ is the free stream Mach number, and γ is the specific heat ratio.

The Euler form of the governing equations is easily obtained by dropping the second order viscous terms [3]. In this instance, the tangential and normal velocity components at solid boundaries must be calculated; this is conveniently handled using simple cosine relations [7].

For incompressible flow, the energy equation reduces to the equation for the scalar transport of temperature. The pressure is obtained from solution of the "discrete" momentum equations and a simple Poisson equation based on the SIMPLE algorithm developed by Patankar [8]. A potential function, which is solved from the Poisson equation, is used to correct the velocity components calculated in the previous time step.

THE FINITE ELEMENT METHOD

Bilinear isoparametric quadrilateral elements are used to discretize two-dimensional problem domains and trilinear hexahedral elements for three-dimensions. The standard weak formulation of the Galerkin weighted residual technique is employed to cast Eqs. (1-5) into their integral form:

Conservation of Mass

$$\int_\Omega\left(\frac{\partial \rho}{\partial t}+\frac{\partial \rho u}{\partial x}+\frac{\partial \rho v}{\partial y}+\frac{\partial \rho w}{\partial z}\right)W_i\,d\Omega=0 \qquad (11)$$

Conservation of Momentum

x-direction

$$\int_{\Omega}\left(\rho\frac{\partial u}{\partial t}+\rho u\frac{\partial u}{\partial x}+\rho v\frac{\partial u}{\partial y}+\rho w\frac{\partial u}{\partial z}+\frac{\partial p}{\partial x}-\frac{\partial \sigma_{xx}}{\partial x}-\frac{\partial \sigma_{xy}}{\partial y}-\frac{\partial \sigma_{xz}}{\partial z}\right)W_i d\Omega=0 \tag{12}$$

y-direction

$$\int_{\Omega}\left(\rho\frac{\partial v}{\partial t}+\rho u\frac{\partial v}{\partial x}+\rho v\frac{\partial v}{\partial y}+\rho w\frac{\partial v}{\partial z}+\frac{\partial p}{\partial y}-\frac{\partial \sigma_{xy}}{\partial x}-\frac{\partial \sigma_{yy}}{\partial y}-\frac{\partial \sigma_{yz}}{\partial z}\right)W_i d\Omega=0 \tag{13}$$

z-direction

$$\int_{\Omega}\left(\rho\frac{\partial w}{\partial t}+\rho u\frac{\partial w}{\partial x}+\rho v\frac{\partial w}{\partial y}+\rho w\frac{\partial w}{\partial z}+\frac{\partial p}{\partial z}-\frac{\partial \sigma_{xz}}{\partial x}-\frac{\partial \sigma_{yz}}{\partial y}-\frac{\partial \sigma_{zz}}{\partial z}\right)W_i d\Omega=0 \tag{14}$$

Conservation of Energy

$$\int_{\Omega}\Big(\frac{\partial E_t}{\partial t}+\frac{\partial (E_t+p)u}{\partial x}+\frac{\partial (E_t+p)v}{\partial y}+\frac{\partial (E_t+p)w}{\partial z}-\frac{\partial}{\partial x}(u\sigma_{xx}+v\sigma_{xy}+w\sigma_{xz}-q_x)$$

$$-\frac{\partial}{\partial y}(u\sigma_{yx}+v\sigma_{yy}+w\sigma_{yz}-q_y)-\frac{\partial}{\partial z}(u\sigma_{zx}+v\sigma_{zy}+w\sigma_{zz}-q_z)\Big)W_i d\Omega=0 \tag{15}$$

where Ω denotes the computational domain and W_i is the weighting function. The ρ, u, v, and E_t variables are represented by the trial approximations

$$\left.\begin{aligned}
\rho(x,y,z,t)&=\sum N_i(x,y,z)\hat{\rho}_i(t)\\
u(x,y,z,t)&=\sum N_i(x,y,z)\hat{u}_i(t)\\
v(x,y,z,t)&=\sum N_i(x,y,z)\hat{v}_i(t)\\
w(x,y,z,t)&=\sum N_i(x,y,z)\hat{w}_i(t)\\
E_t(x,y,z,t)&=\sum N_i(x,y,z)\hat{E}_{t_i}(t)\\
T(x,y,z,t)&=\sum N_i(x,y,z)\hat{T}_i(t)\\
p(x,y,z,t)&=\sum N_i(x,y,z)\hat{p}_i(t)
\end{aligned}\right\} \tag{16}$$

where N_i is the linear basis function; in this instance, $W_i = N_i$. The matrix equivalent formulations of Eqs. (11-15) can be expressed as

$$\dot{M}\underline{\rho}+\bar{\bar{u}}C^T\underline{\rho}+\underline{\rho}C^T\bar{\bar{u}}=0 \tag{17}$$

$$M\dot{\underline{u}}+K\underline{u}+A(\bar{\bar{u}})\underline{u}+C^x p=F_u \tag{18}$$

$$M\dot{\underline{v}}+K\underline{v}+A(\bar{\bar{u}})\underline{v}+C^y p=F_v \tag{19}$$

$$M\dot{\underline{w}}+K\underline{w}+A(\bar{\bar{u}})\underline{w}+C^z p=F_w \tag{20}$$

$$M\dot{\underline{E}}_t + K\underline{E}_t + A(\bar{u})\underline{E}_t = F_E \tag{21}$$

where the $\dot{}$ refers to time differentiation, $\underline{}$ denotes column vector, and $\bar{u}$ is the velocity vector. The matrix coefficients are defined (using integration by parts for the second order viscous terms) as

$$M \equiv m_{i,j} = \int_\Omega N_i N_j d\Omega \tag{22}$$

$$K \equiv k_{i,j} = \int_\Omega \left(\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} + \frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z} \right) d\Omega \tag{23}$$

$$A(\bar{u}) \equiv a_{i,j} = \int_\Omega \left(u_k N_k N_i \frac{\partial N_j}{\partial x} + v_k N_k N_i \frac{\partial N_j}{\partial y} + w_k N_k N_i \frac{\partial N_j}{\partial z} \right) d\Omega \tag{24}$$

$$C^x \equiv c^x_{i,j} = \int_\Omega N_i \frac{\partial N_j}{\partial x} d\Omega \tag{25}$$

$$C^y \equiv c^y_{i,j} = \int_\Omega N_i \frac{\partial N_j}{\partial y} d\Omega \tag{26}$$

$$C^z \equiv c^z_{i,j} = \int_\Omega N_i \frac{\partial N_j}{\partial z} d\Omega \tag{27}$$

$$C^T = \begin{pmatrix} C^x \\ C^y \\ C^z \end{pmatrix} \tag{28}$$

$$F_u = \int_{d\Omega} N_i \left(n_x \frac{\partial u}{\partial x} + n_y \frac{\partial u}{\partial y} + n_z \frac{\partial u}{\partial z} \right) dS_u \tag{29}$$

$$F_v = \int_{d\Omega} N_i \left(n_x \frac{\partial v}{\partial x} + n_y \frac{\partial v}{\partial y} + n_z \frac{\partial v}{\partial z} \right) dS_v \tag{30}$$

$$F_w = \int_{d\Omega} N_i \left(n_x \frac{\partial w}{\partial x} + n_y \frac{\partial w}{\partial y} + n_z \frac{\partial w}{\partial z} \right) dS_w \tag{31}$$

$$F_T = \int_{d\Omega} N_i \left(n_x \frac{\partial T}{\partial x} + n_y \frac{\partial T}{\partial y} + n_z \frac{\partial T}{\partial z} \right) dS_E \tag{32}$$

where the i, j, and k subscripts denote summation over the local nodes within an element and dS_u, dS_v, dS_w, dS_E represent the boundary segments over which u, v, w, and E_t are specified.

Based on the procedure discussed in Pepper and Humphrey [3], the following modifications are employed:

Mass Lumping

Equation (22) is modified to the form

$$M = \delta_{i,j} \int_{\Omega} N_i \, d\Omega \tag{33}$$

where $\delta_{i,j}$ is the Kronecker delta. This operation produces a diagonal matrix in lieu of the sparse global matrix consisting of n x n nodes, thus eliminating the need for implicit matrix solution - hence $[M]^{-1} = 1/M_i$.

Reduced Integration

The value of an integral over an element is obtained at the element centroid using one Gauss point. The shape function becomes

$$N_i = \frac{1}{4} \quad (2-D) \tag{34a}$$

$$N_i = \frac{1}{8} \quad (3-D) \tag{34b}$$

The advection velocities are evaluated at the centroid of an element and multiplied by the average gradient over the element, i.e.,

$$a^x_{i,j} = \tilde{u} \int_{\Omega} N_i \frac{\partial N_j}{\partial x} d\Omega = \tilde{u} c^x_{i,j} \tag{35}$$

$$a^y_{i,j} = \tilde{v} \int_{\Omega} N_i \frac{\partial N_j}{\partial y} d\Omega = \tilde{v} c^y_{i,j} \tag{36}$$

$$a^z_{i,j} = \tilde{w} \int_{\Omega} N_i \frac{\partial N_j}{\partial z} d\Omega = \tilde{w} c^z_{i,j} \tag{37}$$

where

$$A = a^x_{i,j} + a^y_{i,j} + a^z_{i,j}$$

The average velocities are obtained as

$$\tilde{u} = \frac{1}{4}\sum_{i=1}^{4} u_i, \quad \tilde{v} = \frac{1}{4}\sum_{i=1}^{4} v_i \quad (2-D) \tag{38a}$$

$$\tilde{u} = \frac{1}{8}\sum_{i=1}^{8} u_i, \quad \tilde{v} = \frac{1}{8}\sum_{i=1}^{8} v_i, \quad \tilde{w} = \frac{1}{8}\sum_{i=1}^{8} w_i \quad (3-D) \tag{38b}$$

The gradient terms are simple 4x4 matrices in 2-D and 8x8 matrices in 3-D. For example, $c^x_{i,j}$ is defined for the 2-D quadrilateral as

$$c^x_{i,j} = \frac{1}{8}\begin{pmatrix} a & b & -a & -b \\ a & b & -a & -b \\ a & b & -a & -b \\ a & b & -a & -b \end{pmatrix} \tag{39}$$

where a = $y_2 - y_4$ and b=$y_3 - y_1$.

The diffusion terms are evaluated in 2-D as

$$k^x_{i,j} = \int_\Omega \frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} d\Omega = \frac{16}{\Omega_e} c^x_{i,j} c^x_{j,i} \tag{40a}$$

$$k^y_{i,j} = \int_\Omega \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} d\Omega = \frac{16}{\Omega_e} c^y_{i,j} c^y_{j,i} \tag{40b}$$

and in 3-D

$$k^x_{i,j} = \int_\Omega \frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} d\Omega = \frac{64}{\Omega_e} c^x_{i,j} c^x_{j,i} \tag{41a}$$

$$k^y_{i,j} = \int_\Omega \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} d\Omega = \frac{64}{\Omega_e} c^y_{i,j} c^y_{j,i} \tag{41b}$$

$$k^z_{i,j} = \int_\Omega \frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z} d\Omega = \frac{64}{\Omega_e} c^z_{i,j} c^z_{j,i} \tag{41c}$$

where

$$K = k^x_{i,j} + k^y_{i,j} + k^z_{i,j}$$

The gradient terms have been incorporated to take advantage of the $c^x_{i,j}$, $c^y_{i,j}$, and $c^z_{i,j}$ formulations.

Petrov-Galerkin

The weighting functions associated with the Petrov-Galerkin formulation are obtained by perturbing the shape functions such that

$$W_i = N_i + \frac{\alpha \rho h_e}{2\bar{u}}\left(u\frac{\partial N_i}{\partial x} + v\frac{\partial N_i}{\partial y} + w\frac{\partial N_i}{\partial z}\right) \tag{42}$$

where h_e is the element size [3] and α is defined as

$$\alpha = \coth\beta_i - \frac{1}{\beta_i} \qquad i = 1,2 \tag{43}$$

with $\beta_1 = |\bar{u}| h_e R_e/2$, which is the cell Reynolds number; for the energy equation, $\beta_2 = P_r\beta_1$. For the Euler equations, R_e nor P_r appear in the expressions for β. This form of anisotropic balancing diffusion acts in the direction of the propagation of the perturbation (velocity field). The precise amount of artificial diffusion (for eliminating the shortest waves) and direction in which it must be added for optimizing accuracy are calculated for each element.

Time Integration

An explicit forward-in-time Euler scheme is used to advance the discretized equations in time. For example, u is solved from the relation

$$\underline{u}^{n+1} = \underline{u}^n + \Delta t M^{-1}(F_u - K\underline{u}^n - A(\bar{u})\underline{u}^n - C^x p) \tag{44}$$

Application of this scheme to the continuity, v- and w-momentum and energy equations follows similarly. Since the discretized equations are solved explicitly, a stability constraint on the time step must be imposed. The Courant and diffusive limits associated with a forward Euler scheme are calculated over each element, and the time step adjusted to the minimum value within the computational domain.

EXAMPLES

A set of six example problems are solved by the FEM to demonstrate the versatility of the method. Most of the examples were initially run on an IBM PC AT (enhanced with a 33MHz accelerator board [9]) and an Everex 25MHz 386 machine; the first four problems required nodal meshes using less than 2000 nodes. Although problems requiring upwards of 5000 nodes have been successfully run on PCs [2], such jobs may require many hours of calculational effort.

The example problems were also run on an Alliant VFX/40 mini-supercomputer. A Silicon Graphics Personal IRIS workstation served as a graphical front end to the Alliant. This combination of hardware allowed near real time graphical displays of the transient solutions as they converged to steady state. Convergence was assumed to be reached when differences between old and new values (all variables) equalled 10^{-4}.

Example 1:

The first problem deals with fluid recirculation within a square cavity ($0 \le x \le 1; 0 \le y \le 1$), as shown in Fig. 1(a,b). The top surface represents a fluid moving horizontally left to right at u=1, v=0 with R_e = 5000; the remaining three walls are no-slip surfaces.

A non-uniform rectangular mesh of 51x51 nodes is used to model the fluid motion. Ghia, et al [10] and Gresho, et al [4] also analyzed flow in a cavity for a large range of Reynolds numbers. Figure 1(b) shows the streamline distributions, and agrees with their published results. A primary vortex develops within the center region of the cavity; secondary vorticies occur at the two bottom corners and near the upper left corner.

Example 2:

In the second problem, natural convection within a rectangular enclosure is simulated for $R_a=10^5$ and $P_r=1$. A 20x20 nodal mesh is used with zero initial conditions for u, v, and T (Fig. 2(a)).

At $R_a=10^5$, two distinct cells develop from one large single cell shortly after the calculation is begun. This recirculation pattern is consistent with multicellular development as the Rayleigh number increases. Fig. 2(b,c) shows the streamlines and isotherms for steady state conditions, and agree with results in the literature [11]. At higher Rayleigh numbers, a finer grid is required near the walls to properly account for the boundary layer growth.

Example 3:

The ability of the FEM to model convective cooling of a heated block is demonstrated in the third example problem. Prediction of the recirculation zone behind the block and the thermal plume eminating from the

block are examined for R_e=P_e=2000. The problem geometry and mesh are shown in Fig. 3(a,b); the mesh consists of 732 nodes (665 2-D elements).

Since the mesh is coarse, spurious results are expected for R_e=P_e=2000; at lower values, the mesh is adequate. In this case, the ability of the Petrov-Galerkin scheme to control dispersion errors (oscillations in the solution) is assessed. Figure 4(a,b,c,d) shows velocity vectors and isotherms (with and without Petrov-Galerkin weighting). At high Reynolds numbers, the FEM yields poor velocity distributions without Petrov-Galerkin weighting; the temperature distribution shows appreciable dispersion - such solutions generally indicate inadequate mesh refinement. Use of Petrov-Galerkin reduces the oscillations and provides a good solution with a less than optimal mesh.

Example 4:

In the fourth problem, a planar 15° ramp sits on the lower wall of a two-dimensional duct. The inlet Mach number is 2.28. Based on 1-D ideal analysis (weak solution for attached shock waves), an oblique shock develops at an angle of 40° relative to the freestream. Figure 5(a,b,c) shows the flow schematic and problem geometry. The shock reflects off the top wall at an angle of 40.5°. In region 2, M_2 = 1.69 and p_2/p_1 (static) = 2.34. In region 3, M_3 = 1.15 and p_3/p_1 = 4.91.

Figure 6(a,b) shows density contours and velocity vectors for the viscous solution. A time step of $\Delta t = 10^{-4}$ is initially required. The boundary layer development is apparent; at the upper wall, the reflected shock interacts with the boundary layer and creates a small recirculation zone. For this viscous solution, R_e = 10^4; turbulence has not been included in the model as yet. Although the shock is smeared, the shock angle and downstream Mach numbers agree with the 1-D analysis. A more optimized grid refinement (more uniformity and nodes) is required to reduce spreading of the shock. Using 2x2 Gauss points versus one Gauss point increases accuracy, but also increases the compute time.

Example 5:

The fifth problem is concerned with supersonic flow over a projectile within a two-dimensional chamber. This problem is similar to the Scramaccelerator work conducted by Pratt, et al [12] and Humphrey [13]; the projectile is designed so that the reflected shock off the tube wall initiates combustion (oblique detonation wave), thereby producing an increase in pressure behind the shocks and a net thrust forward. In this example, the upper wall moves at u=1 (non-dimensional). Approach flows for Mach 2.68 (nose angle 20°) and Mach 5 (nose angle 25°) are simulated with the initial flow field assumed to be uniform (i.e., equal to the inlet conditions) throughout the domain. The problem geometry and mesh are shown in Fig. 7.

Figure 8(a,b,c,d) shows pressure and Mach contours for Mach 2.68 and Mach 5 inlet conditions, respectively. An oblique bow shock forms over the nosë and reflects off the top wall. High pressure gradients develop in the region between the wall and the projectile. Figure 9(a,b) shows an expanded view of the velocity vectors in front of the projectile. In Fig. 9(a), boundary layer separation occurs on the upper part of the nose as a result of the interaction with the reflected shock; in this

case, the inlet flow speed is not high enough and the shock inhibits forward motion. At Mach 5 and a 25° leading edge (Fig. 9b), the oblique shock reflects off the upper wall and strikes the projectile just behind the shoulder (near optimal performance). A weak shock train is created in the thin region between the upper wall and projectile body; boundary layer separation occurs along the top of the projectile body. The predicted oblique shock angles and downstream flow conditions near the front shoulder of the projectile agree quantitatively with 2-D results obtained by Pratt, et al [12] using a modified SALE algorithm [14].

Example 6:

For the last problem, 3-D flow is calculated around a set of heated obstacles which are convectively cooled by cold air. The physical domain and mesh are shown in Fig. 10(a,b). The mesh consists of 2868 hexahedral elements (which easily permits execution on a Personal IRIS workstation); the Reynolds number is $R_e=10^3$ and $P_r=1.0$. This type of problem is commonly encountered in the computer industry where cooling of computer chips is critical.

Figure 11(a,b) gives normal and perspective views of the 3-D velocity vectors within the channel. Recirculation of the flow occurs behind the blocks, and small secondary cells develop in the corners. The pressure and isotherm distributions are shown in Fig. 12(a,b) for the x-y plane near the top of the channel. Distinct thermal plumes eminate from the heated blocks; plume impingement from the left forward block occurs on the small mid-stream block. It is well known that when flow separates at the corners of blocks, horseshoe-like vortices are generated [15]. Further work is underway to examine this phenomena using a 28,000 hexahedral element mesh on the Alliant.

CONCLUSIONS

A finite element method using simple modifications is used to solve 2- and 3-D problems for compressible and incompressible flows with heat transfer. The modified FEM employs equal order basis functions for all unknown variables, mass lumping, reduced quadrature, and Petrov-Galerkin weighting; transient solutions are solved using an explicit Euler scheme. Formulation of the local (and global) matrices are simple, and solution speeds are quick. The technique yields accurate results for a wide variety of flows providing the elements are not too distorted. The method appears attractive for workstation class machines.

ACKNOWLEDGEMENTS

We wish to thank Mr. Andy Singer, Advanced Projects Research, Inc., for his help in generating many of the graphical displays, and Dr. Juan Heinrich and Mr. Frank Brueckner, University of Arizona, Tucson, Arizona, for their assistance in generating the Scramaccelerator results.

REFERENCES

1. Thompson, J. F., "Numerical Solution of Flow Problems using Body Fitted Coordinate Systems," *Computational Fluid Dynamics*, 1, (W. Kollman, ed.), Hemisphere Pub. Co., Washington, D.C. (1980).

2. Pepper, D. W. and Singer, A. P., "Calculation of Convective Flow on the Personal Computer using a Modified Finite Element Method," to appear in Num. Heat Transfer (1990).

3. Pepper, D. W. and Humphrey, J. W., "A Hybrid Finite Element Method for Compressible Flow," AIAA Paper 90-0399 (1990).

4. Gresho, P. M., Chan, S. T., Lee, R. L., and Upson, C. D., "Modified Finite Element Method for Solving the Time-Dependent Incompressible Navier-Stokes Equations, Part 1: Theory," Int. J. Num. Methods in Fluids, 4, 557-598 (1984).

5. Yu, C. C. and Heinrich, J. C., "Petrov-Galerkin Methods for the Time-Dependent Convective Transport Equation," Int. J. Num. Methods in Eng., 23, 883-901 (1986).

6. Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, Hemisphere Pub. Co., Washington, D. C. (1984).

7. Shapiro, R. A., "An Adaptive Finite Element Solution Algorithm for the Euler Equations," MIT Report CFDL-TR-88-7 (1988).

8. Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere Pub. Co., Washington, D.C. (1980).

9. Definicon Systems, Inc., "PM Series Product Manual," Definicon Systems, Inc., 1100 Business Center Circle #5, Newbury Park, CA 91320 (1989).

10. Ghia, U., Ghia, K., and Shin, C., "High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multi-grid Method," J. Comp. Phys., 48, 387-401 (1982).

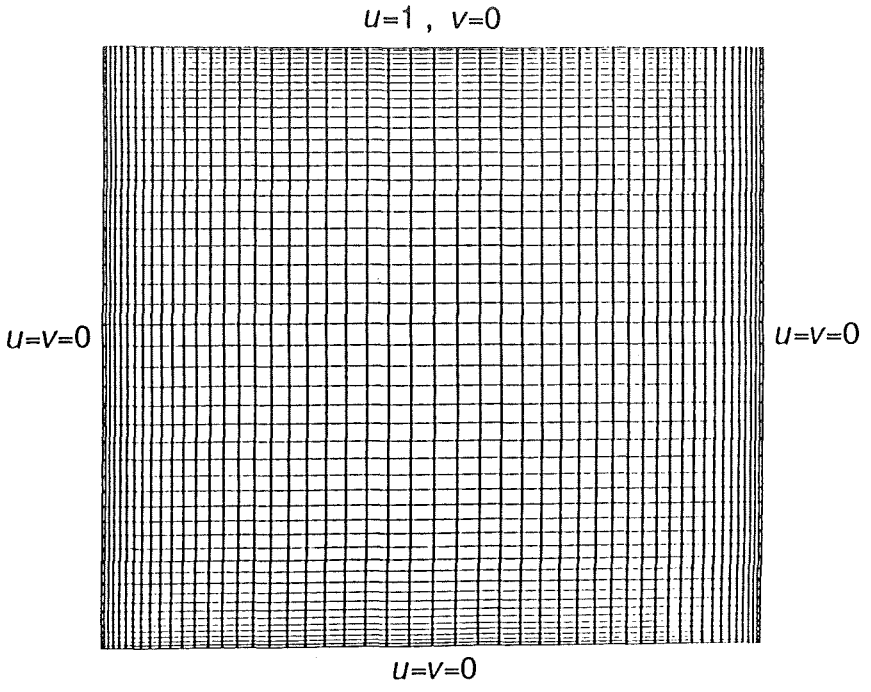
11. Jones, I. P. and Thompson, C. P. (ed.), "Numerical Solutions for a Comparison Problem on Natural Convection in an Enclosed Cavity," AERE-R 9955, Harwell, England (1980).

12. Pratt, D. T., Humphrey, J. W., and Glenn, D. E., "Morphology of a Standing Oblique Detonation Wave," AIAA Paper 87-1785 (1987).

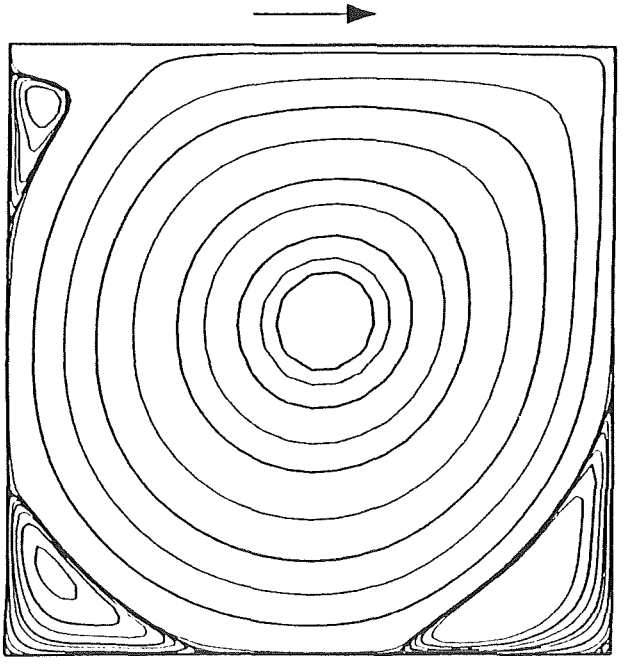
13. Humphrey, J. W., "Study of an Oblique Detonation Wave Ramaccelerator Driven Hypersonic Test Facility," APRI Final Report, NASA Langley Contract NAS1-18802 (1989).

14. Amsden, A. A., Ruppel, H. M., and Hirt, C. W., "SALE: A Simplified Arbitrary Lagrangian-Eulerian Computer Program for Fluid Flow at All Speeds," Los Alamos Nat. Lab. Report LA-8095, UC-32 (1980).

15. Hunt, J. C., Abell, C. J., Peterka, J. A., and Woo, H., "Kinematical Studies of the Flows around Free or Surface Mounted Obstacles; Applying Topology to Flow Visualization," J. Fluid Mech., 86, 1, 179-200 (1978).

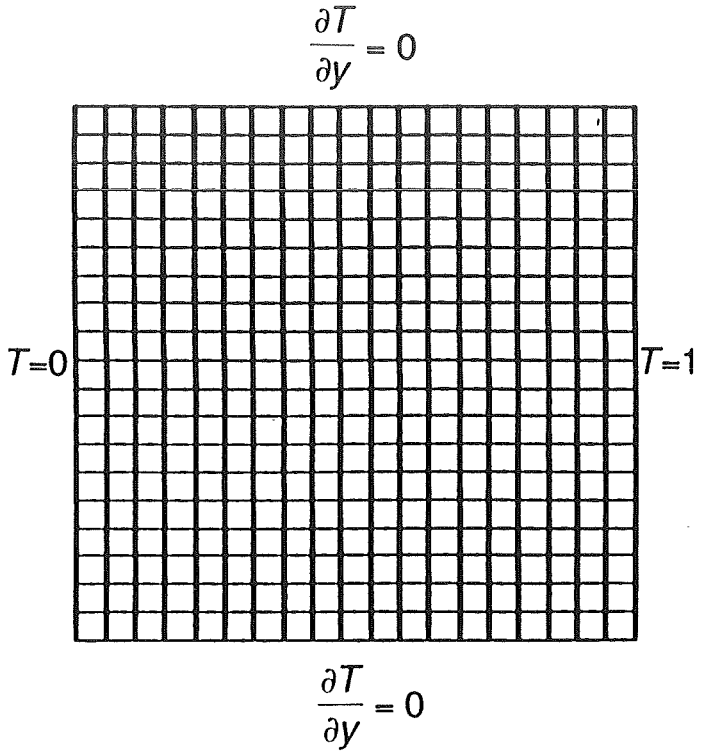


a.) Mesh

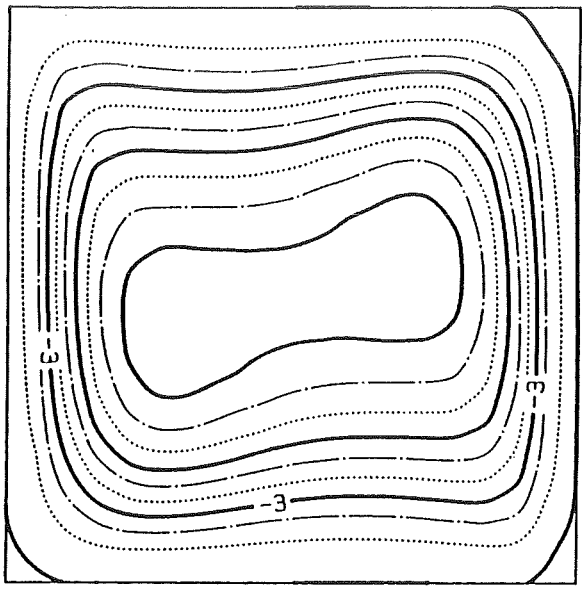


b.) Streamlines

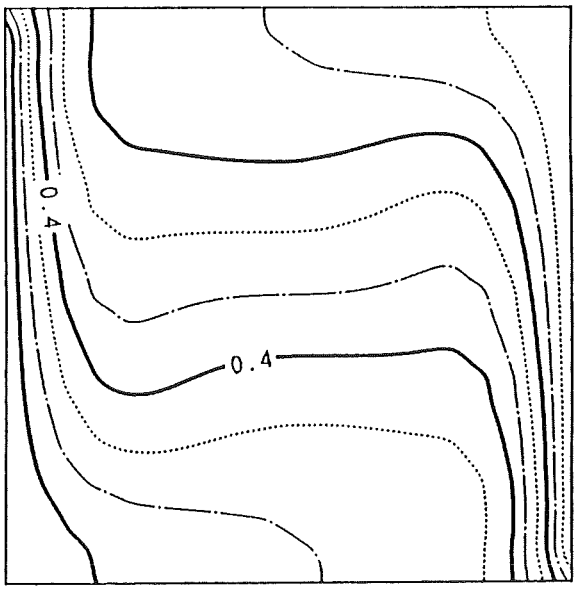
Figure 1. Flow in a Cavity - Re=5000



a.) Mesh

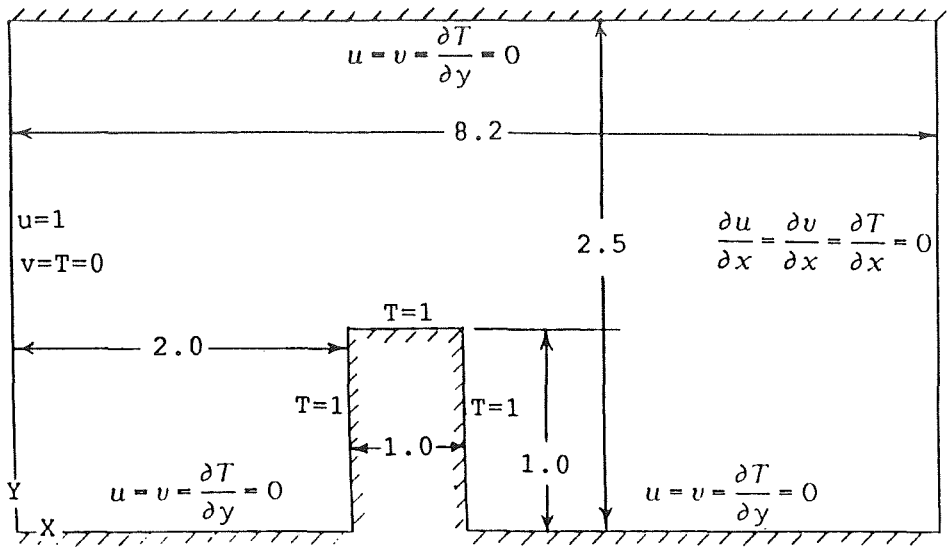


b.) Streamlines

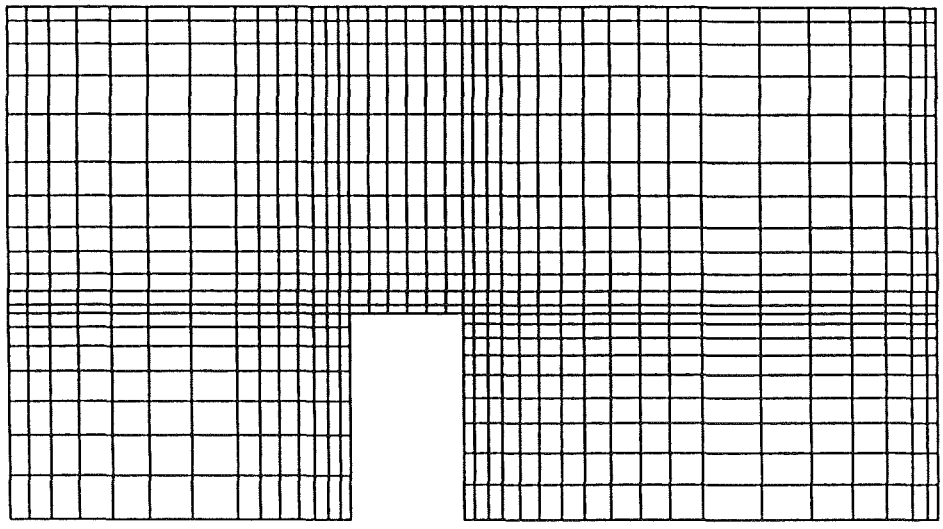


c.) Isotherms

Figure 2. Natural Convection - Ra = 10^5

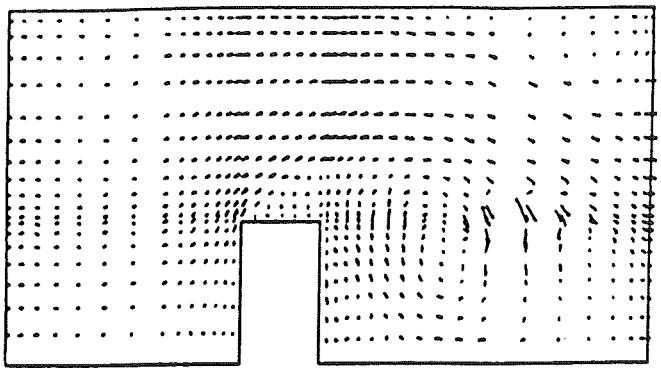


a.) Boundary Conditions

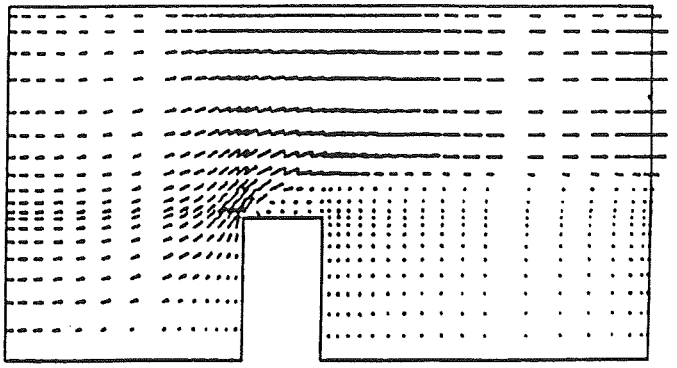


b.) Mesh

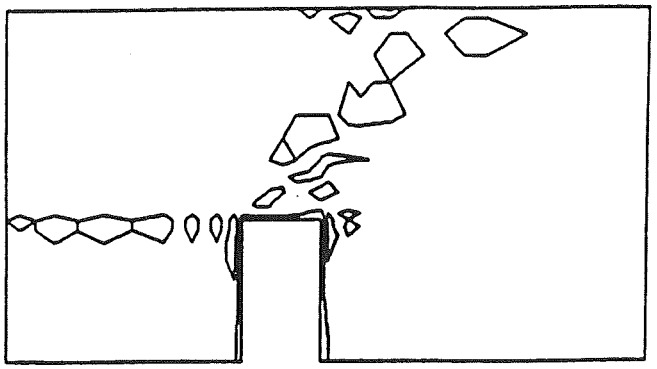
Figure 3. Forced Convective Cooling - Re=Pe=2000



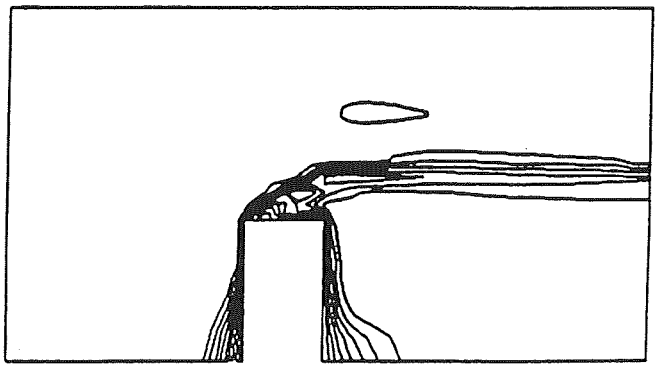
a.) w/o Petrov-Galerkin Weighting



c.) Velocity Vectors
(w/ Petrov-Galerkin Weighting)

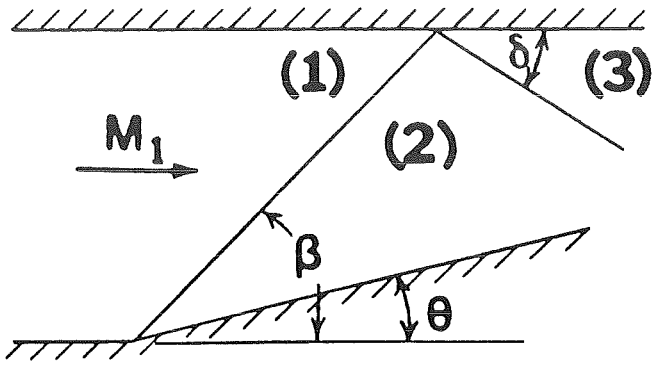


b.) Temperature Distribution
(w/o Petrov-Galerkin Weighting)

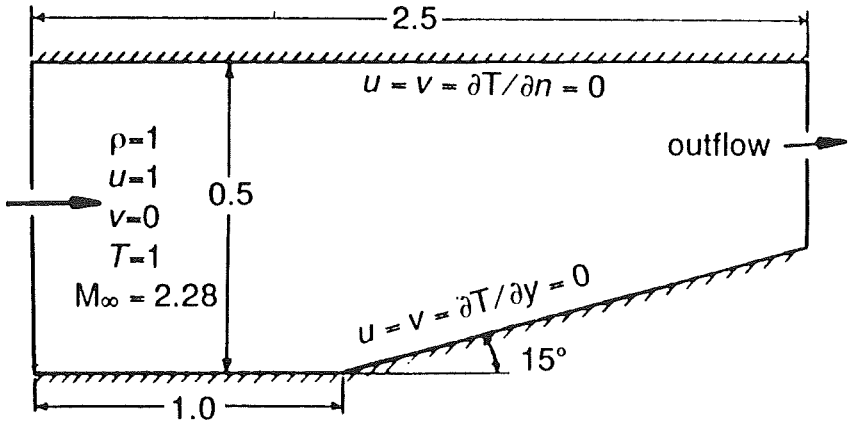


d.) Temperature Distribution
(w/ Petrov-Galerkin Weighting

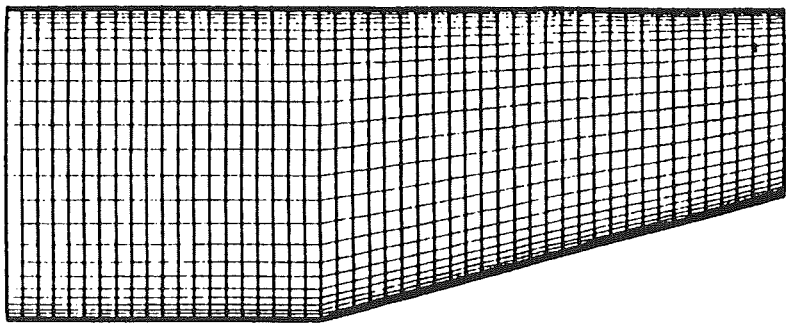
Figure 4. Forced Convective Cooling - Velocity Vector and Isotherms



a.) Flow schematic for 15° Ramp
(0 = 15°; δ = 40.5°;β = 40°)

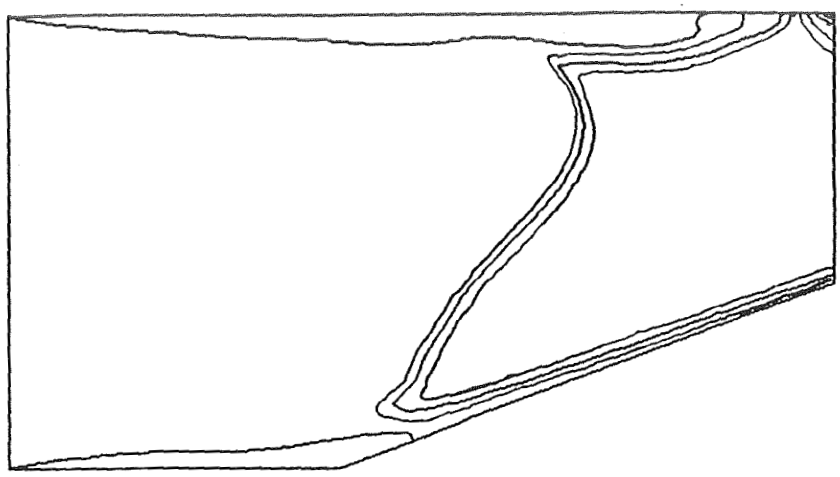


b.) Problem domain

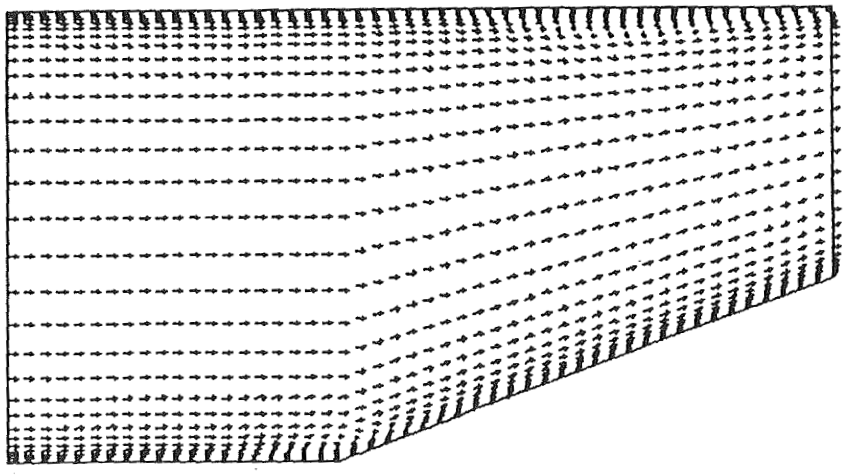


c.) Mesh for viscous solution

Figure 5. Supersonic flow over a 15° Ramp



a.) Density contours



b.) Velocity vectors

Figure 6. Supersonic Flow over a 15° Ramp - Viscous Solution

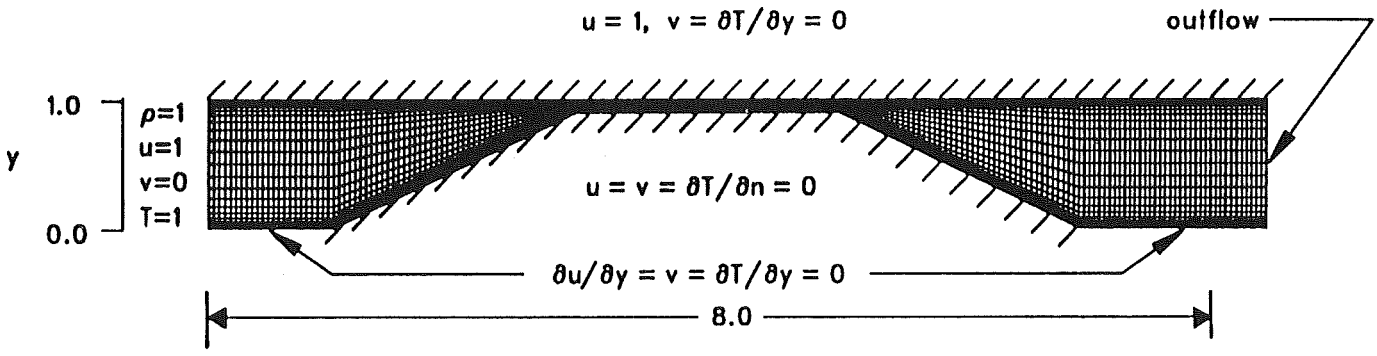
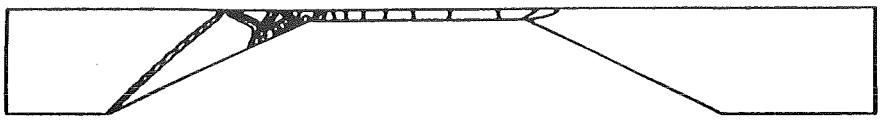


Figure 7. Scramaccelerator Configuration

$M_\infty = 2.68$

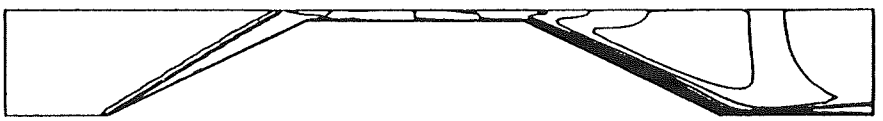


$M_\infty = 5.00$

a.) Pressure Contours

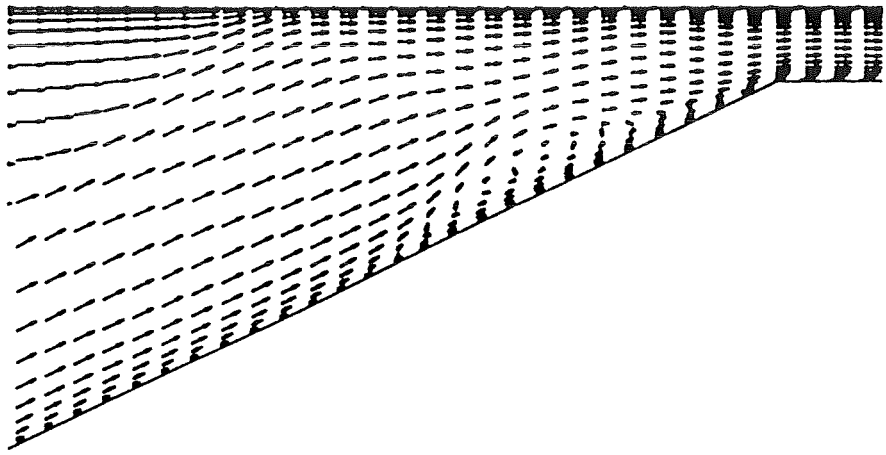
$M_\infty = 2.68$

$M_\infty = 5.00$

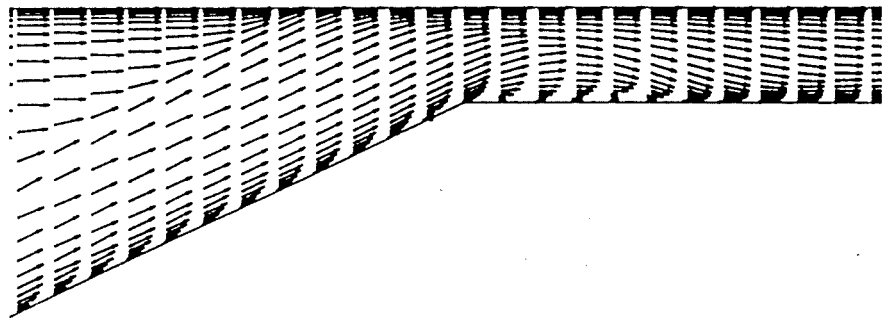


b.) Mach Contours

Figure 8. Scramaccelerator

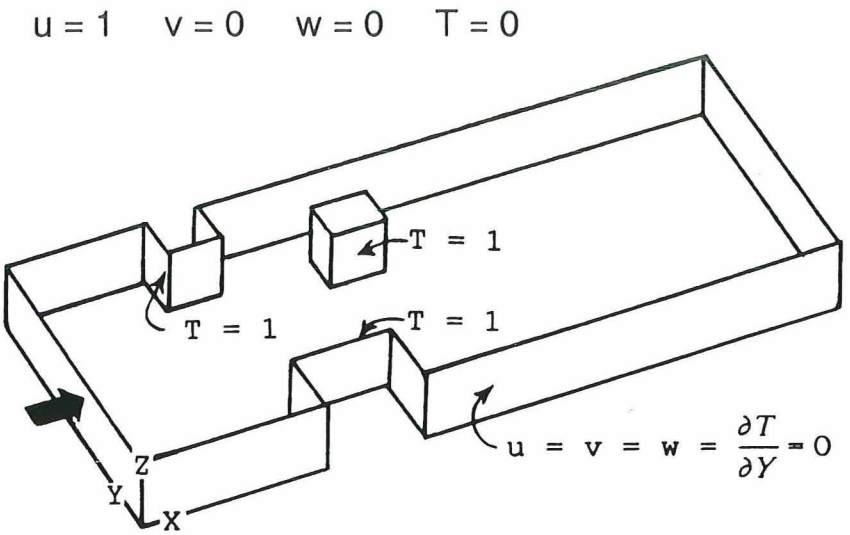


$M_\infty = 2.68$

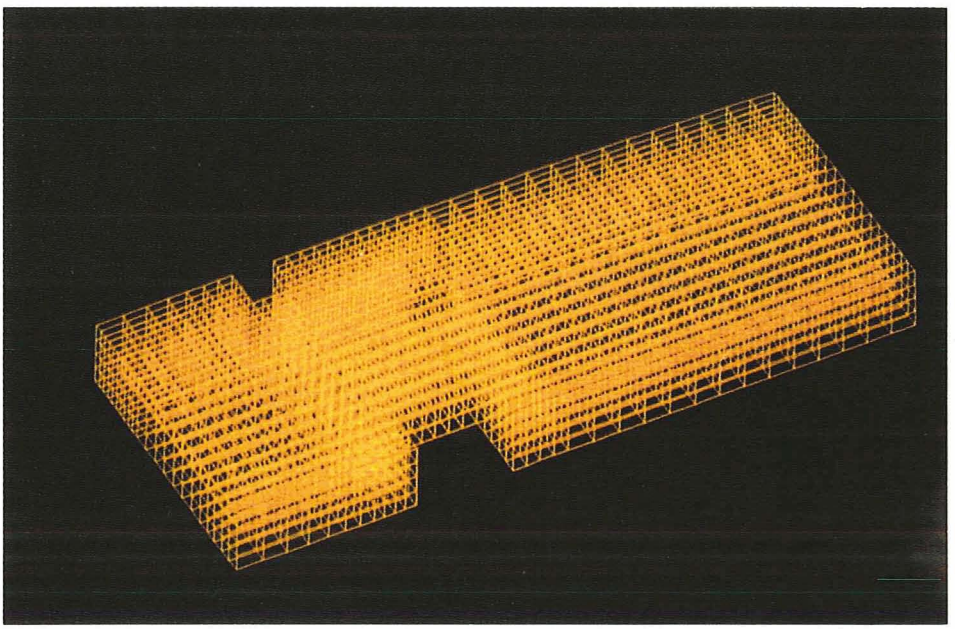


$M_\infty = 5.00$

Figure 9. Inlet Velocity Distribution - Scramaccelerator

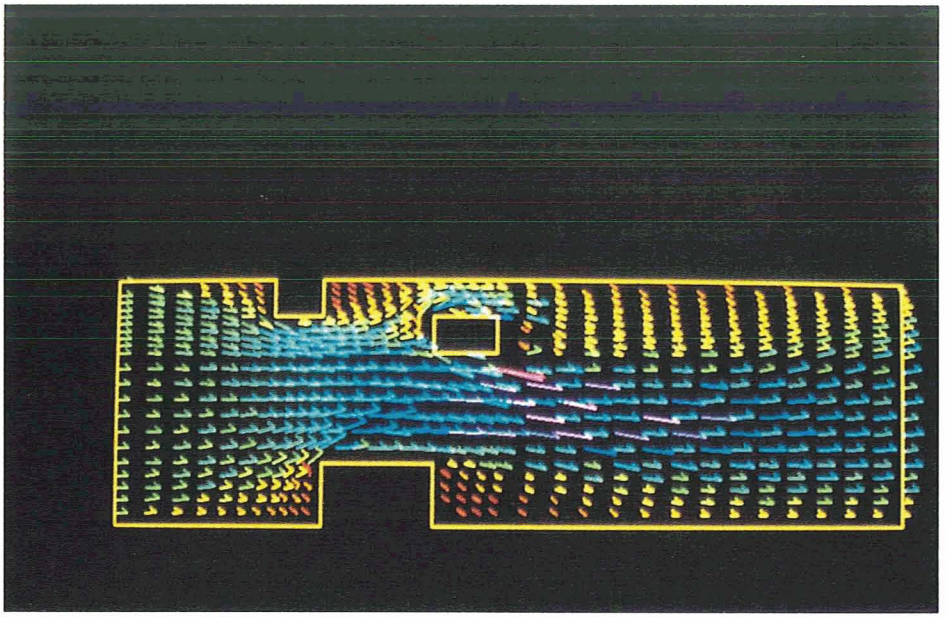


a.) Boundary Conditions

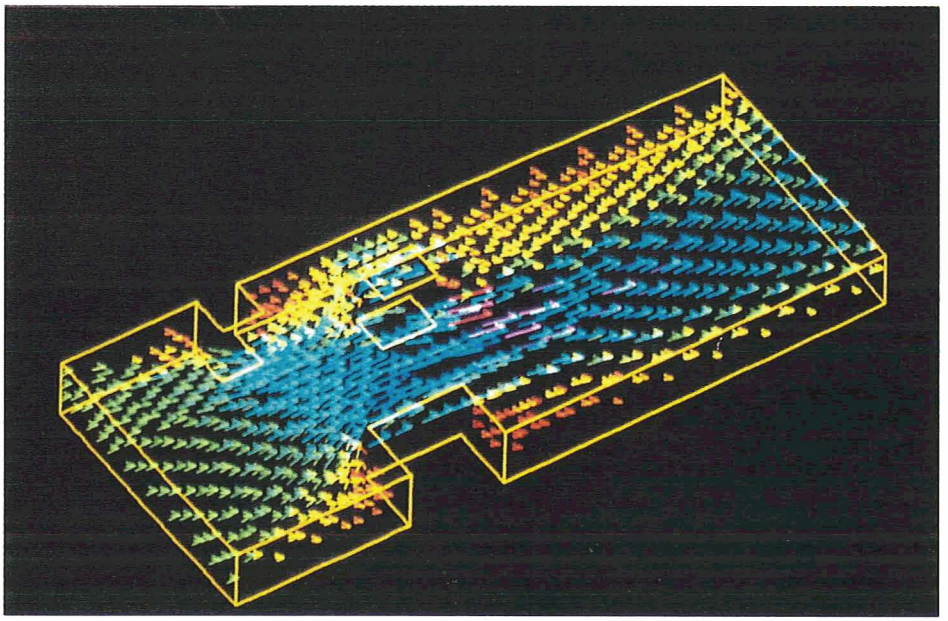


b.) Mesh

Figure 10. Channel with Obstacles - Forced convective Flow

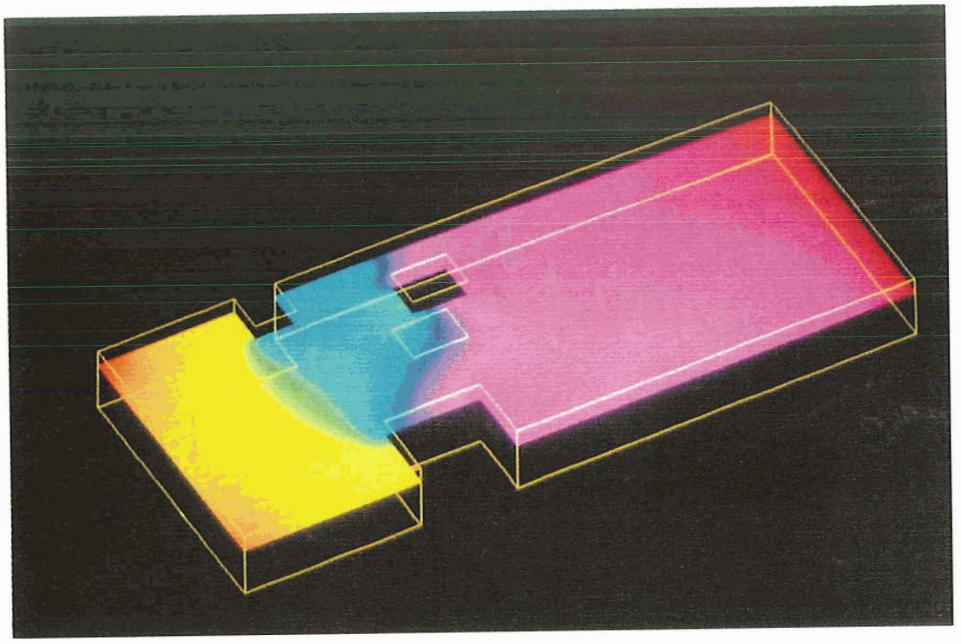


a.) View from Above

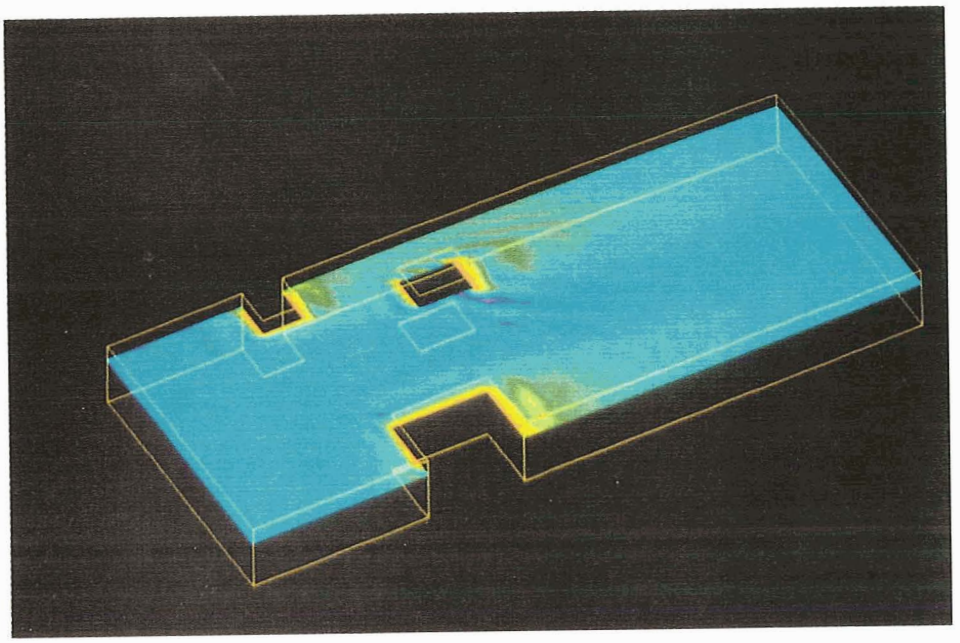


b.) Perspective View

Figure 11. Velocity Distribution



a.) Pressure



b.) Isotherms

Figure 12. distribution in X-Y Plane